SIGEVOlution

newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation

Volume 5
Issue 3

in this issue

An Interview with
Stewart W. Wilson
with an introduction by Martin V. Butz

Reassembling
Operator
Equalisation
Sara Silva

GECCO-2011
best paper awards

calls & calendar

Editorial

It is a great pleasure and a great honor for me to introduce this new issue of SIGEVOlution that hosts an interview with Stewart W. Wilson. To me, Stewart is a mentor and a great friend. To our community, he is the person who, with XCS and XCSF, singlehandedly revolutionized learning classifier system research. In 1994, his paper on XCS in the Evolutionary Computation Journal, brought new life to an almost stalled research field, providing the community with the first classifier system that could tackle a wide variety of difficult problems. In 2001, he introduced XCSF a classifier system for function approximation and opened up a completely new horizon. If you are doing research in learning classifier system today, you are probably using one of his models or something derived from them.

The second paper by Sara Silva presents a new recipe for operator equalization to control bloat in Genetic Programming. The paper has been one of the best paper nominees at GECCO-2011 and is reprinted here with the permission of ACM and Sara. An extended version of it also appears in the new Genetic Programming Theory and Practice volume.

FOGA-2013 is looking for a home! The deadline to propose a venue to host such an interesting event is September 30 so hurry up! You can find the detailed call for proposal inside the newsletter.

I hope you like the cover!

Pier Luca
September 22, 2011

SIGEVOlution Volume 5, Issue 3

Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation.

Contributors to this Issue

Stewart W. Wilson
Martin V. Butz
Sara Silva

Contents

An Interview with Stewart W. Wilson **2**
Martin V. Butz
Call for Proposals to Host and Chair FOGA-2013 **7**
GECCO-2011 Best Paper Awards **8**
Reassembling Operator Equalisation: A Secret Revealed **10**
Sara Silva
Calls and Calendar **23**
About the Newsletter **29**

ISSN: 1931-8499

An Interview with Stewart W. Wilson

with an introduction by Martin V. Butz

Stewart W. Wilson, Prediction Dynamics, Concord, MA 01742, wilson@prediction-dynamics.com

Stewart W. Wilson is certainly one of the most innovative and functionality-oriented thinkers I have ever had the honor to meet. His research career commenced at the Massachusetts Institute of Technology (MIT), where he received an S.B. degree in physics and an S.M. and PhD degree in electrical engineering. After having completed his formal education, research work at the Polaroid Corporation (starting in 1962) and the Rowland Institute for Science (starting in 1983) led him to the development of practically useful, yet highly complex, biologically-inspired systems and machine learning architectures. In 1998 he founded the research and consultancy company Prediction Dynamics. Since 1999 he has been an adjunct professor at the University of Illinois at Urbana-Champaign, IL.

Albeit Learning Classifier Systems were proposed by John H. Holland in the 1970s, Stewart is most closely associated with these systems. As the inventor of the zeroth-level classifier system ZCS in 1994 and the accuracy-based classifier system XCS in 1995, he has set a standard in this research realm that is still valid and highly useful today. In fact, the XCS classifier system may be considered one of the most powerful genetics-based machine learning tools available. Originally designed to solve Boolean function problems and maze-like reinforcement learning problems, XCS has now been successfully applied to various other problem domains. In datamining, XCS has shown very robust and highly competitive classification results and has also generated insights into the underlying problem structure. In online generalizing reinforcement learning problems, XCS has successfully solved problems with very huge state spaces (more than 10^{27} states). In function approximation problems, XCS has solved high-dimensional regression problems with near optimal final solution distributions. Finally, in the robotics domain, XCS has been shown to autonomously learn forward and inverse kinematic models that are highly useful for the flexible, task-dependent control of redundant actuator systems. Thus, Stewart not only created a good system for solving Boolean functions, but he actually initiated a whole machine learning branch in the form of a very flexible, highly innovative learning architecture.

In addition to his intellectual achievements, Stewart has always been a great friend, mentor, and research partner to me and many others. Having gotten the opportunity to talk with him over many years, I can only wish to be and stay as open minded, approachable, and innovative as he.

Martin V. Butz, University of Würzburg

Q Everybody knows the enormous influence you had in our field. Would you summarize the key ideas of learning classifier systems in 2-3 paragraphs for someone unfamiliar with the field?

The most key idea is Holland's proposal that you could base a learning system on an evolving population of condition-action rules. The rules (classifiers) would compete and/or cooperate to obtain maximum payoff (reinforcement) from the environment. They would be evolved based on their ability to get payoff, resulting in a system containing rules that better and better fit the environment, including an environment that changed with time.

His second key idea was to propose a solution for environments that did not provide payoff upon each action of the system, but only did so at the end of a sequence of actions. His bucket-brigade technique built on an idea of Samuel's for checkers, and in effect passed payoff back to classifiers that set the stage for later payoff-achieving actions. The bucket brigade helped inspire the field of reinforcement learning and was one of its first examples.

A third key idea has to do with how the fitness of a classifier is measured. Originally the amount of payoff a classifier received determined its fitness. However, it was discovered that basing fitness on the accuracy of a classifier's prediction of payoff– instead of on the payoff itself–resulted in much better performance and led to the first systems that worked robustly and reliably. They also generalized accurately over environmental regularities.

A fourth key idea is the introduction of classifiers that compute their payoff prediction instead of simply asserting a scalar value. The computation is usually a linear function of the input vector, but can be higher-order as well. This leads to stronger generalization ability as well as use of the system as an adaptive function approximator.

Q

What experiences in school, if any, influenced you to pursue a career in science?

In secondary school I was interested in nearly every subject and all my teachers were good, with the English and French teachers probably the most influential. However, in math and science I liked being able to be more sure of something than in other subjects. I was also interested in what lay behind things, including the universe— e.g., "Why is the sun?", asked the 15-year-old. So when the opportunity came to go to Harvard or MIT, I picked MIT. Later, I became increasingly interested in intelligence, learning, and "what we are".

Q

Who are the three people whose work inspired you the most in your research?

Chronologically, Edwin H. Land, Jerome Lettvin, and John Holland.

Land invented sheet polarizers, created instant photography and the Polaroid Corporation, and introduced the first significantly non-Newtonian theory of color vision. He was a creative genius who knew how to challenge people to do their very best, and he understood that failure had to be a part of it. He taught me about what it means to be a scientist, which I will illustrate with a couple of quotations:

> "If I don't have one good experiment a day,
> the world tends to go out of focus".
>
> "If the experiment doesn't work, distrust the
> experiment.
> If the experiment works, distrust the theory."
>
> "Science is a method to keep
> yourself from fooling yourself".

Jerome Lettvin was a pioneer of neuroscience ("What the frog's eye tells the frog's brain"). He constantly asked what could be there that we don't yet see, that may be bigger than we imagine. Jerry was very helpful to me when I was thinking about vision. He pointed me toward Helmholz's description of the peripheral vision experiment of Aubert and Foerster, which deeply affected how I thought about vision and the brain/mind.

With John Holland, as with Land and Lettvin, I was inspired by his constant thinking outside–beyond–the mainstream. When in about 1980 I discovered John's book in the basement of the MIT library, apparently never having been checked out, I was immediately set on my path to classifier systems.

Here at last was somebody asking the right questions about learning and adaptation, which I believed had been ignored by AI, and he asked those questions precisely and from a perspective that seemed olympian in its scope.

When shortly afterwards I met John at an Ann Arbor seminar he kindly invited me to, I was attracted by his unusual combination of intellectual openness and faith in his own ideas.

Q

What are the three books or papers that inspired you most?

One paper, “Pandemonium: a paradigm for learning” (1959), by Oliver Selfridge, had an early effect on me. I knew Oliver, who was very smart and constantly thought "outside of the box".

I would mention also the book “Brains, Behavior, & Robotics”, by James S. Albus. In it he introduced the very clever CMAC learning algorithm which he derived from observations of the cerebellum. Albus also has much to say about hierarchy in cognition and learning.

Finally, I would mention John Koza’s first book, which always impressed me with its sudden creation of another evolutionary mainstream.

Q

As a founding father of this field, what is your own view about what learning classifier systems are? What did you expect them to be?

Well, I would say they are a powerful and versatile cognitive model. Of course, the model is still fairly simple. But I think its strength compared with other architectures such as neural networks is that it is based on rules and is open in the sense that a computation in one part of the system (population) can be independent of other computations going on.

This gives the possibility–if we are clever– of adaptively building computational levels or hierarchies, of having one part of the system observe other parts, etc.

What did I expect? Just this kind of thing!

Q

What do you like most about EC?

The magic. And the people.

Q

What do you dislike most about EC?

Only seeing it, as other fields sometimes do, get too involved in details. In this respect, a venture like the “Humies” competition is wonderful.

Q

What is the biggest open question in the evolutionary computation area?

How far it can be taken. I.e., what is the maximum level of problem difficulty or complexity that can be solved by evolutionary methods? What does this depend on: adequate representation, algorithmic techniques, hardware?

Q

Where do you see the evolutionary computation community going in the next ten years? Twenty years?

I can’t comment beyond what I have already said.

What are your favorite real-world applications of learning classifier systems?

One was some work for a client on an important oil industry problem–the ability to instantaneously measure the actual flow of oil in a pipe. This is hard because usually the oil is mixed up with water, sand and gravel, air and other gases. The classifier system received an input derived from an acoustic signal, i.e., from "listening" to the pipe. Amazingly, the system (XCSF in this case) learned to output the correct actual oil flow. This is an example of learning in a complex and nonlinear situation with no known analytical solution.

There are many other applications, especially in data mining, autonomous robotics, and control. Larry Bull's book "Applications of Learning Classifier Systems" is a good source of examples.

Your papers are sources of inspirations. Is there any topic in your papers which you hoped people would take more seriously?

Hmm. I've been lucky in that most topics have been taken up and extended eventually. However, several could be pushed further.

One is internal state in a classifier system–which permits the system to learn in non-Markov environments. This is fundamental to getting intelligent robotics. Another is pushing the ability of individual classifiers to generalize over environmental regularities, thus reducing the number of classifiers needed and increasing readability (perhaps by the system itself) of the knowledge. Generalization would seem to depend on increasing the syntactic flexibility of classifier conditions.

My most recent paper is the outcome of several years' thinking about pattern recognition. It takes a new and rather unexpected viewpoint which I hope others will want to follow up.

Which ones are the most misunderstood/misquoted?

Here again, I've been lucky.

If you could do it again, what would you do differently in your development of the evolutionary computation field?

I don't have any regrets about paths not taken, only that I might have had more energy or been smarter!

What new ideas are you working on and excited about?

Those of the recent paper just mentioned in which a co-evolutionary situation is set up in which programs are induced to generate and recognize increasingly sophisticated patterns—in an effort to allow exploration of operators and methods without the limitations of human preconceptions as to what they should be.

In another theme, I am investigating prediction from time-based data streams where the histories are quite limited, so that the identification of significant features is paramount. This seems characteristic of life itself, and has many interesting applications.

What books, tangentially related to the field, that you've read in the last year did you like the best?

I finally read "Atlas Shrugged" and greatly enjoyed it. Besides being a terrific story, it describes a titanic contest of libertarian and anti-libertarian or socialist forces that recalls some of evolutionary computation, at least metaphorically.

Q

You had many successful PhD students. What is your recipe for PhD success?

Pick an inspiring advisor who will remember what you are doing but will let you find your own way. Choose a problem or topic that you believe is very important, for which you believe you have a special insight or perspective, and which with enough effort you know you can solve or contribute to in a major way.

Document your experiments and thinking so you can reproduce them, both for others and for yourself.

Q

Your key advice to a PhD student?

Don't give up! If you do you will always regret it.

Q

What advice would you give to students and beginning researchers who are starting to work in evolutionary computation?

From reading and talking to people, find a nascent idea that you think is valuable but needs much further development. Pick it up and run with it.

Q

Has thinking about evolution changed your view on things in general?

There is no doubt that evolutionary computation has given me a clearer picture of natural evolution, even though EC is in a sense only a sketch. Has thinking about evolution affected my view in general? It has increased my appreciation for things like survival-oriented motivations, as well as the wonders of genetic possibility. I would say, though, that I resist going to an exclusively evolutionary world-view just yet, because I think there is still much more to be learned.

About the author

Stewart W. Wilson was born in Rochester, NY, USA. In 1960, he received the S.B. degree in physics from MIT. He received the S.M. and Ph.D. degrees in electrical engineering from MIT in 1962 and 1967. His research and consulting entity is Prediction Dynamics, Concord, MA. He was associated at Polaroid Corporation with Dr. Edwin H. Land in investigations of systems that allowed students to learn via asking their own questions of well-known scientists. Later, at the Rowland Institute for Science, Cambridge, MA, he continued his long-term interest in computer programs that learn, with special focus on the classifier systems that had been introduced by John H. Holland. Dr. Wilson is an Adjunct Professor in the Department of Department of Industial and Enterprise Systems Engineering of the University of Illinois at Urbana/Champaign. He is an Associate in VGO Associates, the systems consulting firm founded and headed by David Davis. He is on the Advisory Board of *Evolutionary Computation* and is a member of the Editorial Boards of *Artificial Life* and *Adaptive Behavior*. He is a co-founder of *Adaptive Behavior* and the Simulation of Adaptive Behavior (SAB) conferences. Besides learning systems and perception, he is interested in history and politics (free market views), and classical music.

Homepage: http://www.prediction-dynamics.com
Email: wilson@prediction-dynamics.com

Call For Proposals to Host and Chair FOGA

The “Foundations of Genetic Algorithms" (FOGA) meetings have been held approximately every two years since 1990. In that time, FOGA has expanded to cover all aspects of the theoretical foundations of all Evolutionary Algorithms.

Proposals are being soliciting to host and chair FOGA sometime during 2012 or 2013. Proposals should include information on where FOGA would be held, and background information should be provided about the proposed organizers. Proposals should also include proposed dates for FOGA.

Proposals should be submitted to Darrell Whitley whitley@CS.ColoState.EDU and Wolfgang Banzhaf banzhaf@mun.ca no later than September 30, 2011. The final selection will be made by the SIGEVO Executive Committee and proposers will be notified by October 17, 2011.

FOGA is 100% sponsored by the Association for Computing Machinery (ACM) Special Interest Group on Genetic and Evolutionary Computation, SIGEVO.

Important Dates

Proposal Submission	September 30, 2011
Notification	October 17, 2011

GECCO-2011 Best Paper Awards

In 2002, ISGEC created a best paper award for GECCO. As part of the double blind peer review, the reviewers were asked to nominate papers for best paper awards. We continue the tradition this year. The Track Chairs, Editor in Chief, and the Conference Chair nominated the papers that received the most nominations and/or the highest evaluation scores for consideration by the conference. The winners are chosen by secret ballot of the GECCO attendees after the papers have been orally presented at the conference. Best Paper winners are posted on the conference website. The titles and authors of all the best papers awarded at GECCO-2011 are given below:

Ant Colony Optimization and Swarm Intelligence

An Incremental ACOR with Local Search for Continuous Optimization Problems. Tianjun Liao (IRIDIA, CoDE, Universite Libre de Bruxelles), Marco Montes de Oca (IRIDIA, CoDE, Universite Libre de Bruxelles), Dogan Aydin (Ege University), Thomas Stützle (IRIDIA, CoDE, Universite Libre de Bruxelles), Marco Dorigo (IRIDIA, CoDE, Universite Libre de Bruxelles)

Artificial Life/Robotics/Evolvable Hardware

Spontaneous Evolution of Structural Modularity in Robot Neural Network Controllers. Josh Bongard (University of Vermont)

Bioinformatics, Computational, Systems, and Synthetic Biology

A Genetic Algorithm to Enhance Transmembrane Helices Topology Prediction Using Compositional Index. Nizar Zaki (UAE University), Salah Bouktif (UAE University), Sanja Molnar (UAE University)

Digital Entertainment Technologies and Arts

Interactively Evolving Harmonies through Functional Scaffolding. Amy Hoover (University of Central Florida), Paul Szerlip (University of Central Florida), Kenneth Stanley (University of Central Florida)

Evolutionary Combinatorial Optimization and Metaheuristics

A Cooperative Tree-based Hybrid GA-B&B Approach for Solving Challenging Permutation-based Problems. Malika Mehdi (University of Luxembourg & INRIA Lille), Jean-Claude Charr (INRIA Lille Nord-Europe - University of Lille), Nouredine Melab (INRIA Lille Nord-Europe - University of Lille), EL-Ghazali Talbi (INRIA Lille Nord-Europe - University of Lille), Pascal Bouvry (University of Luxembourg)

Estimation of Distribution Algorithms

Hierarchical Allelic Pairwise Independent Functions. David Iclănzan (Sapientia Hungaryan University of Transylvania)

Evolutionary Multiobjective Optimization

Improved S-CDAS using Crossover Controlling the Number of Crossed Genes for Many-objective Optimization. Hiroyuki Sato (The University of Electro-Communications), Hernan Aguirre (Shinshu University), Kiyoshi Tanaka (Shinshu University)

Evolution Strategies and Evolutionary Programming

Local-Meta-Model CMA-ES for Partially Separable Functions. Zyed Bouzarkouna (IFP Energies nouvelles), Anne Auger (INRIA), Didier Yu Ding (IFP Energies nouvelles)

Genetic Algorithms

How Crossover Helps in Pseudo-Boolean Optimization. Timo Kötzing (Max-Planck-Institute for Informatics), Dirk Sudholt (University of Birmingham), Madeleine Theile (Technische Universität Berlin)

Genetics Based Machine Learning

Modelling the Initialisation Stage of the ALKR Representation for Discrete Domains and GABIL Encoding. Maria Franco (University of Nottingham), Natalio Krasnogor (University of Nottingham), Jaume Bacardit (University of Nottingham)

Genetic Programming

Rethinking Multilevel Selection in Genetic Programming. Shelly Wu (Memorial University of Newfoundland), Wolfgang Banzhaf (Memorial University of Newfoundland)

Generative and Developmental Systems

On the Relationships between Synaptic Plasticity and Generative Systems. Paul Tonelli (ISIR, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222), Jean-Baptiste Mouret (ISIR, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222)

Real World Applications

RankDE: Learning a Ranking Function for Information Retrieval using Differential Evolution. Danushka Bollegala (The University of Tokyo), Nasimul Noman (The University of Tokyo), Hitoshi Iba (The University of Tokyo)

Search-Based Software Engineering

Searching for Invariants using Genetic Programming and Mutation Testing. Sam Ratcliffe (University of York), David White (University of York), John Clark (University of York)

Self-* Search

Policy Matrix Evolution for Generation of Heuristics. Ender Ozcan (University of Nottingham), Andrew Parkes (University of Nottingham)

Theory

An Analysis on Recombination in Multi-Objective Evolutionary Optimization. Chao Qian (Nanjing University), Yang Yu (Nanjing University), Zhihua Zhou (Nanjing University)

Reassembling Operator Equalisation: A Secret Revealed

Sara Silva — INESC-ID Lisboa, IST / UNL, Portugal — CISUC, University of Coimbra, Portugal — sara@kdbio.inesc-id.pt

The recent Crossover Bias theory has shown that bloat in Genetic Programming can be caused by the proliferation of small unfit individuals in the population. Inspired by this theory, Operator Equalisation is the most recent and successful bloat control method available. In this work we revisit two bloat control methods, the old Brood Recombination and the newer Dynamic Limits, hypothesizing that together they contain the two main ingredients that make Operator Equalisation so successful. We reassemble Operator Equalisation by joining these two ingredients in a hybrid method, and test it in a hard real world regression problem. The results are surprising. Operator Equalisation and the hybrid variants exhibit completely different behaviors, and an unexpected feature of Operator Equalisation is revealed, one that may be the true responsible for its success: a nearly flat length distribution target. We support this finding with additional results, and discuss its implications.

1 Introduction

The most recent theory concerning bloat is the Crossover Bias theory introduced by Dignum, Poli and Langdon [11, 5, 6]. It explains code growth in tree based GP by the effect that standard subtree crossover has on the distribution of tree sizes, or program lengths, in the population. Whenever subtree crossover is applied, the amount of genetic material removed from the first parent is the exact same amount inserted in the second parent, and vice versa. The mean tree size, or mean program length, remains unchanged.

However, as the population undergoes repeated crossover operations, it approaches a particular Lagrange distribution of tree sizes where small individuals are much more frequent than the larger ones [6]. For example, crossover generates a high amount of single-node individuals. Because very small individuals are generally unfit, selection tends to reject them in favor of the larger individuals, causing an increase in mean tree size. According to the theory, it is the proliferation of these small unfit individuals, perpetuated by crossover, that ultimately causes bloat. Strong theoretical and empirical evidence supports the Crossover Bias theory. It has been shown that the bias towards smaller individuals is more intense when the population mean tree size is low, and that the initial populations resembling the Lagrange distribution bloat more easily than the ones initialized with traditional methods [11]. It was also found that the usage of size limits may actually speed code growth in the early stages of the run, as it promotes the proliferation of the smaller individuals [6]. Along with further theoretical developments, it has also been shown that smaller populations bloat more slowly [14], and that elitism reduces bloat [13, 12].

Inspired by the Crossover Bias theory, Operator Equalisation [7, 17] is the most recent and successful bloat control method available today. It can bias the population towards a desired program length distribution by accepting or rejecting each newly created individual into the population. Operator Equalisation can easily avoid the small unfit individuals resulting from the crossover bias, as well as the excessively large individuals that are no better than the smaller ones already found.

Preventing the larger individuals from entering the population is a common bloat control practice; preventing the smaller ones is not, however it has been done non explicitly. We revisit two bloat control methods, the old Brood Recombination [21] and the newer Dynamic Limits [16], hypothesizing that together they contain these two key ingredients that seem to make Operator Equalisation so successful. We reassemble Operator Equalisation by joining them in a hybrid method, and test it in a hard real world regression problem, revealing surprising results.

In the next section we describe Operator Equalisation with some detail. Sections 3 and 4 describe Brood Recombination and Dynamic Limits, explaining why they contain the necessary ingredients to reassemble Operator Equalisation. Section 5 describes the data, techniques and parameters used for the experiments, while Section 6 reports and discusses all the results obtained. Finally, Section 7 summarizes and draws conclusions, also suggesting future work.

2 Operator Equalisation

Developed alongside the Crossover Bias theory (see Section 1), Operator Equalisation is a recent technique to control bloat that allows an accurate control of the program length distribution inside a population during a GP run. Already used a number of times in benchmark and real world problems (e.g. [18, 19, 22, 20]), it is however still fairly new, so we provide a detailed explanation of how it works.

2.1 Distribution of program lengths

We use the concept of a histogram. Each bar of the histogram can be imagined as a bin containing those programs (individuals, solutions) whose length is within a certain interval. The width of the bar determines the range of lengths that fall into this bin, and the height specifies the number of programs allowed within. We call the former *bin width* and the latter *bin capacity*. All bins are the same width, placed adjacently with no overlapping. Each length value, *l*, belongs to one and only one bin *b*, identified as follows:

$$b = \left\lfloor \frac{l-1}{bin\_width} \right\rfloor + 1 \qquad (1)$$

For instance, if $bin\_width = 5$, bin 1 will hold programs of lengths 1,..,5, bin 2 will hold programs of lengths 6,..,10, etc. The set of bins represents the distribution of program lengths in the population.

Operator Equalisation biases the population towards a desired target distribution by accepting or rejecting each newly created individual into the population (and into its corresponding bin). The original idea of Operator Equalisation [7], where the user was required to specify the target distribution and maximum program length, rapidly evolved to a self adapting implementation [17] we here designate as OpEq, where both these elements are automatically set and dynamically updated to provide the best setting for each stage of the evolutionary process. Other developments of Operator Equalisation were also made [18] but we do not use them here.

There are two tasks involved in OpEq: calculating the target (in practical terms, defining the capacity of each bin) and making the population follow it (making sure the individuals in the population fill the set of bins).

2.2 Calculating the Target Distribution

In OpEq the dynamic target length distribution simply follows fitness. For each bin, the average fitness of the individuals within is calculated, and the target is proportional to these values. Bins with better average fitness will have higher capacity, because that is where search is proving to be more successful. Formalizing, the capacity, or target number of individuals, for each bin b, is calculated as:

$$bin\_capacity_b = round(n \times (\bar{f}_b / \sum_i \bar{f}_i)) \qquad (2)$$

where $\bar{f}_i$ is the average fitness in the bin with index i, $\bar{f}_b$ is the average fitness of the individuals in b, and n is the number of individuals in the population. Equation 2 is used for maximization problems where higher fitness is better (so the fitness values must suffer a transformation for minimization problems, for example a sign inversion and mapping back to positive values).

Initially based on the first randomly created population, the target is updated at each generation, always based on the fitness measurements of the current population. This creates a fast moving bias towards the areas of the search space where the fittest programs are, avoiding the small unfit individuals resulting from the crossover bias, as well as the excessively large individuals that are no better than the smaller ones already found. Thus the dynamic target is capable of self adapting to any problem and any stage of the run.

2.3 Following the Target Distribution

In OpEq every newly created individual must be validated before eventually entering the population, and the ones who do not fit the target are rejected. Each offspring is created by genetic operators as in any other GP system. After that, its length is measured and its corresponding bin is identified using Equation 1. If this bin already exists and is not full (meaning that its capacity is higher than the number of individuals already there), the new individual is immediately accepted. If the bin still does not exist (meaning it lies outside the current target boundaries) the fitness of the individual is measured and, in case we are in the presence of the new best-of-run (the individual with best fitness found so far), the bin is created to accept the new individual, becoming immediately full. Any other non-existing bins between the new bin and the target boundaries also become available with capacity for only one individual each. The dynamic creation of new bins frees OpEq from the fixed maximum program length that was present in the original idea. The criterion of creating new bins whenever needed to accommodate the new best-of-run individual is inspired by the Dynamic Limits bloat control technique [16].

When the intended bin exists but is already at its full capacity, or when the intended bin does not exist and the new individual is not the best-of-run, the individual is evaluated and, if we are in the presence of the new best-of-bin (meaning the individual has better fitness than any other already in that bin), the bin is forced to increase its capacity and accept the individual. Otherwise, the individual is rejected. Permitting the addition of individuals beyond the bin capacity allows a clever overriding of the target distribution, by further biasing the population towards the lengths where the search is having a higher degree of success. In the second case, when the bin does not exist and the individual is not the best-of-run, rejection always occurs.

3 Brood Recombination

Brood Recombination, also called Greedy Recombination, was popularized by Tackett in 1994 [21] as a new recombination operator to serve as a substitute for the standard subtree crossover. Instead of recombining two parents once to produce one pair of offspring, Brood Recombination recombines two parents n times, each time selecting different crossover points, to produce n pairs of offspring, where n is called the *brood size factor*. Then only two offspring are selected, the best of the brood, and the rest discarded. This idea was originally introduced by Altenberg as Soft Brood Selection [1], to which Tackett added the use of a reduced-cost fitness evaluation for members of the brood. The primary motivation for developing Brood Recombination was to improve the efficiency of GP systems:

> *"The fitness evaluation of brood members is performed with a 'culling function' which is a fractional subset of the fitness evaluation function for full-fledged population members. A significant result is that large reductions in the cost of the culling function produce small performance degradation of the population members."* [21]

The secondary motivation was to reduce bloat, based on the early and long lasting theory that bloat emerges as a protection against the destructive effects of crossover (e.g. [1, 4, 8, 10], for a review of bloat theories see [16]). But Tackett refutes this theory based on the fact that Brood Recombination, being a much less destructive recombination operator, was not able to reduce code growth. However, according to the Crossover Bias theory, Brood Recombination should help control bloat. In practical terms, creating several pairs of offspring and then choosing only the best may reduce the crossover bias to create many small individuals. If it is verified that the smaller offspring are indeed the most unfit, they will not be selected from among the brood members, and not introduced into the population. The larger the brood, the larger the reduction of bias.

Therefore, we designate Brood Recombination as the first key element for assembling a hybrid method that recreates the successful behavior of Operator Equalisation. We do not, however, use the "culling function" for brood member selection, instead using the same fitness function used for full-fledged population members.

This means we are in fact using the original Soft Brood Selection [1], however we decide to keep the most popular name of Brood Recombination. We also introduce a variant of Brood Recombination which we call Batch Recombination. The only difference is that, instead of repeatedly selecting two offspring from the $2n$ brood members produced by each single couple, all the offspring needed to form a new generation are now selected only once from among the several broods produced by all the couples. This should reduce the crossover bias ever further.

4 Dynamic Limits

Tree-based GP traditionally uses a static depth limit to avoid excessive growth of its individuals. When an individual is created that violates this limit, one of its parents is chosen for the new generation instead [9].

> *"This effectively avoids the growth of trees beyond a certain point, but it does nothing to control bloat until the limit is reached. The static nature of the limit may also prevent the optimal solution to be found for problems of unsuspected high complexity."* [16]

These unsolved problems lead Silva et al. to create a bloat control technique called Dynamic Maximum Tree Depth [15, 16]. It also imposes a depth limit on the individuals accepted into the population, but this one is dynamic, meaning that it can be changed during the run. The dynamic limit is initially set with a low value, usually the same as the maximum depth of the initial random trees. Any new individual who breaks this limit is rejected and replaced by one of its parents (as with the traditional static limit), unless it is the best individual found so far. In this case, the dynamic limit is raised to match the depth of the new best-of-run and allow it into the population. Dynamic Maximum Tree Depth can coexist with the traditional depth limit.

First published in 2003 [15], the original Dynamic Maximum Tree Depth was then extended to include two variants: a heavy dynamic limit, called heavy because it falls back to lower values whenever allowed, and a dynamic limit on size instead of depth. The entire concept has later been collectively designated as Dynamic Limits [16]. The heavy limit is one that accompanies the depth of the best individual, up or down, with the sole constraint of not going lower than its initialization value; a very heavy option allows it to fall back even below its initialization value. As expected, whenever the limit falls back to a lower value, some individuals already in the population immediately break the new limit. These are allowed to remain in the population but, when breeding, the limit that applies to their children is the depth of the deepest parent. The second variation is the usage of a dynamic size limit, where size is the number of nodes of the tree. The dynamic size limit also includes a modified version of the Ramped Half-and-Half initialization procedure that replaces the concept of depth with the concept of size.

Since Operator Equalisation itself was inspired by the Dynamic Limits for the decision on when to open new bins (see Section 2.3), it is only natural to assume that this is the second key element for its success. Although a size limit would probably mimic the decisions made by Operator Equalisation more accurately (because they are based on solution length, which is exactly the same thing), this variant was never as successful as using depth [16] and has the additional burden of a modified initialization procedure, so we decided to use the dynamic limit on depth. We have, however, chosen the very heavy option that allows the limit to fall back as much as possible, since in Operator Equalisation it is also possible to eliminate any bins from the target, in case they remain empty.

5 Experiments

To perform our experiments we chose to use the first real world problem that was tackled by Operator Equalisation, the prediction of the human oral bioavailability of a set of candidate drug compounds on the basis of their molecular structure [18, 19]. We briefly describe the problem and then specify how Operator Equalisation was reassembled using Brood Recombination and Dynamic Limits, specifying the techniques and parameters used in the experiments.

5.1 Test Problem

Human oral bioavailability is the pharmacokinetic parameter that measures the percentage of the initial orally submitted drug dose that effectively reaches the systemic blood circulation after passing through the liver. This parameter is particularly relevant in the drug discovery process, and this problem has already been approached by several machine learning methods, with GP providing the best results so far [2, 3].

We have used the same dataset as [18, 19], which is freely available[1]. The dataset consists of a matrix composed by 260 rows and 242 columns, where each row is a vector of molecular descriptors of a particular drug, and each column represents a molecular descriptor, except the last one that contains the known target values of the bioavailability parameter.

Following [18, 19], from this dataset training and test sets were obtained by random splitting: at each different run, 70% of the molecules were randomly selected with uniform probability and inserted into the training set, while the remaining 30% were used for the test set.

5.2 Techniques and Parameters

To reassemble Operator Equalisation we began by implementing a Standard GP system (StdGP). Then we joined Brood Recombination (Brood), alternatively Batch Recombination (Batch), using different Brood/Batch sizes (2,5,10). To assess how much the Brood/Batch Recombination differs from Standard GP, and how much it pushes the behavior towards Operator Equalisation (OpEq), we compared all of them to each other.

On a second phase we joined Dynamic Limits (Dyn) to all the previous variants to create the hybrid techniques, and once again performed comparisons among them, to assess how much they are able to approximate the behavior of OpEq. Finally we implemented OpEq with a flat target distribution (FlatOpEq) to verify some of our results.

Table 1 shows the numbers and acronyms of the 16 different techniques used. Some of the plots of Section 6 use the numbers for lack of space for the acronyms. Table 2 shows the parameter settings common to all the techniques. Regarding the parameters specific to each technique, both Operator Equalisation techniques use a bin width of 1, and none uses the maximum depth limit.

6 Results and Discussion

Some of the plots presented in this section are still somewhat unconventional. They plot the evolution of fitness against length of the solution, completely disregarding generations, evaluations, or time spent in the search process. These plots have been first used by Silva and Dignum [17] and we consider them to be an intuitive way of visualizing the bloating behavior of any given technique.

Indeed, we are not interested in measuring the performance of the techniques in terms of how much computational effort is required to achieve a given fitness. Operator Equalisation is recognized to be inefficient, an issue discussed at length in [17], however it can find solutions with a fitness/length ratio that other techniques do not seem to be able to reach. In the real world this is usually one of the most important quality factor of a solution, regardless of the more or less lengthy search process that ultimately found it. Therefore, we also present a few plots showing the evolution of length along the generations, knowing perfectly well that one generation represents enormously different computational efforts to different techniques.

Although the issue of overfitting is not central to this work, we are using a real world problem where the generalization ability is important. Therefore, we present some results obtained in the test set, to show that none of the modified techniques suffers from a decreased generalization ability that would prevent it from being successfully used in the real world.

Finally, in most plots and related text we do not discriminate between the different Brood/Batch sizes except when we consider the differences to be important.

6.1 Comparing fitness and solution length

Figure 1a shows the best training fitness plotted against the average length of the solutions in the population, for Standard GP, the different sizes of Brood/Batch Recombination, and Operator Equalisation. Figure 1b is similar to 1a except that instead of plotting the best training fitness, it plots the test fitness of the best training individual. It is immediately apparent that, although some variants of Brood/Batch Recombination exhibit a more desirable bloating behavior (fitness/length ratio) than Standard GP, most of the differences seem to be caused by the simple fact that producing more offspring allows for more search, since the general trend is the same. Also Operator Equalisation is allowed more search due to the number of rejected individuals, however its behavior is not even remotely approximated by any of the Brood/Batch variants.

[1] http://personal.disco.unimib.it/Vanneschi/bioavailability.txt

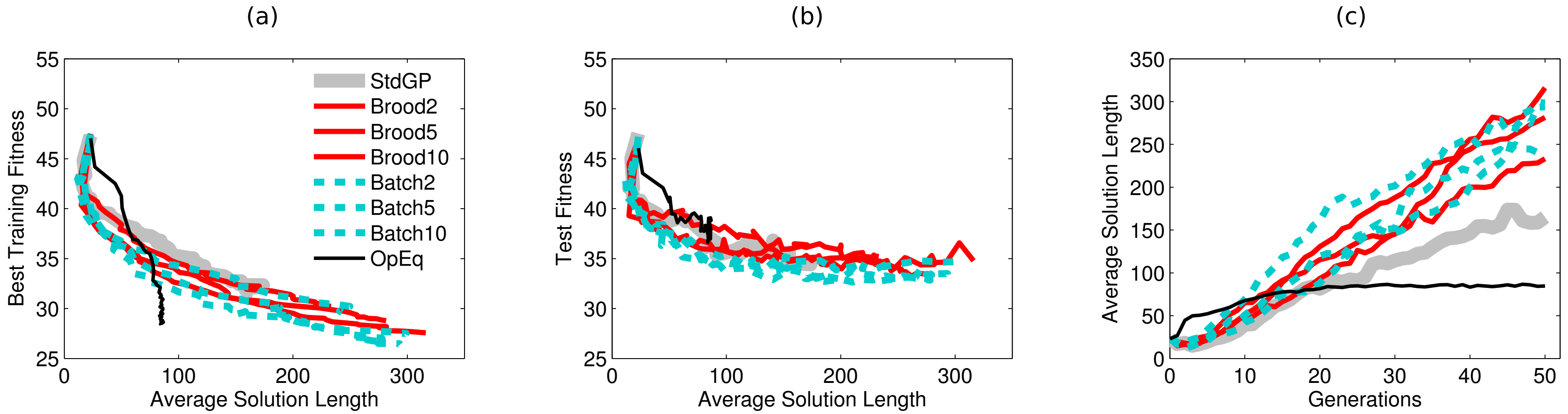


Fig. 1: Standard GP versus Brood/Batch Recombination versus Operator Equalisation. (a) Best training fitness versus average solution length; (b) Test fitness (of the best training individual) versus average solution length; (c) Average solution length versus generations.

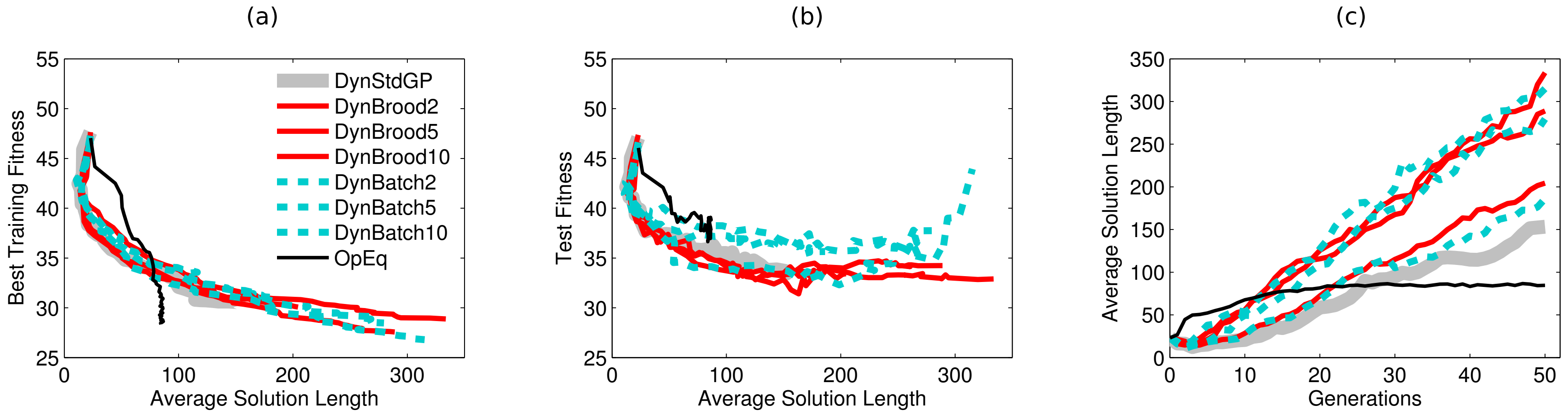


Fig. 2: Dynamic Standard GP versus Dynamic Brood/Batch Recombination versus Operator Equalisation. (a) Best training fitness versus average solution length; (b) Test fitness (of the best training individual) versus average solution length; (c) Average solution length versus generations.

The stabilization of average solution length exhibited by Operator Equalisation had already been observed in other real world problems (e.g. [22, 20]). Figure 1c shows the evolution of the average solution length plotted against the generations, where the differences in code growth are more easily observed. For a visualization of the distribution of training and test fitness and average solution length in the final generation consult Figure 5.

Figure 2 is analogous to Figure 1 except that it refers to Standard GP and Brood/Batch Recombination using Dynamic Limits. Like Brood/Batch Recombination, Dynamic Limits is a somewhat helpful modification by itself (see Figure 4 for a direct comparison), but the effect of joining both elements is not cumulative, and once again Operator Equalisation remains a far better technique in terms of bloating behavior. The usage of Dynamic Limits in Brood/Batch Recombination causes larger differences in the behavior of different Brood/Batch sizes, with size 2 (DynBrood2 and DynBatch2) resulting in less code growth and less learning (see also Figure 5), which may simply be a result of less search.

6.2 Exploring the length distributions

Given the previous plots we are forced to conclude that the hybrid techniques using Brood/Batch Recombination and Dynamic Limits do not contain the same ingredients as Operator Equalisation. By design, the hybrid techniques should emulate the decisions of Operator Equalisation on whether to accept or reject the individuals (large or small) that fall outside the limits of the target, so the difference must lie within the target itself. Therefore, we now focus our attention on the distribution of solution lengths during the evolution, looking for an explanation to the unique behavior of Operator Equalisation. In principle, given the same limits the solution lengths should follow similar distributions in all the techniques, since Operator Equalisation enforces a distribution that is “proportional” to fitness (see Section 2.2), which is exactly what selection is supposed to do.

Figure 3 contains some actual and target length distributions of different techniques. The three plots in the first row (a,b,c) show typical length distributions obtained by Standard GP, Dynamic Brood of size 2, and Operator Equalisation. The height of the peaks is not important for this discussion. DynBrood2 was chosen for being the hybrid technique with the lowest expected difference to Standard GP, and it is interesting to compare the length distributions of both.

Tab. 1: Numbers and acronyms of the 16 techniques used.

Number	Acronym	Number	Acronym
1	StdGP	9	DynBrood2
2	DynStdGP	10	DynBrood5
3	Brood2	11	DynBrood10
4	Brood5	12	DynBatch2
5	Brood10	13	DynBatch5
6	Batch2	14	DynBatch10
7	Batch5	15	OpEq
8	Batch10	16	FlatOpEq

Tab. 2: Parameter settings common to all techniques.

Parameter	Setting
Number of runs	30
Population size	500
Function and terminal sets	$\{+,-,*,/\}$, $\{x_1,...,x_{241}\}$
Tree initialization and growth	ramped max depth 6, limit 17
Fitness function	root mean squared error
Selection for reproduction	lexicographic tournament, size 10
Replication rate	0.1
Genetic operators	crossover 0.9, mutation 0.1
Selection for survival	non elitist, replace all
Stop criterion	50 generations

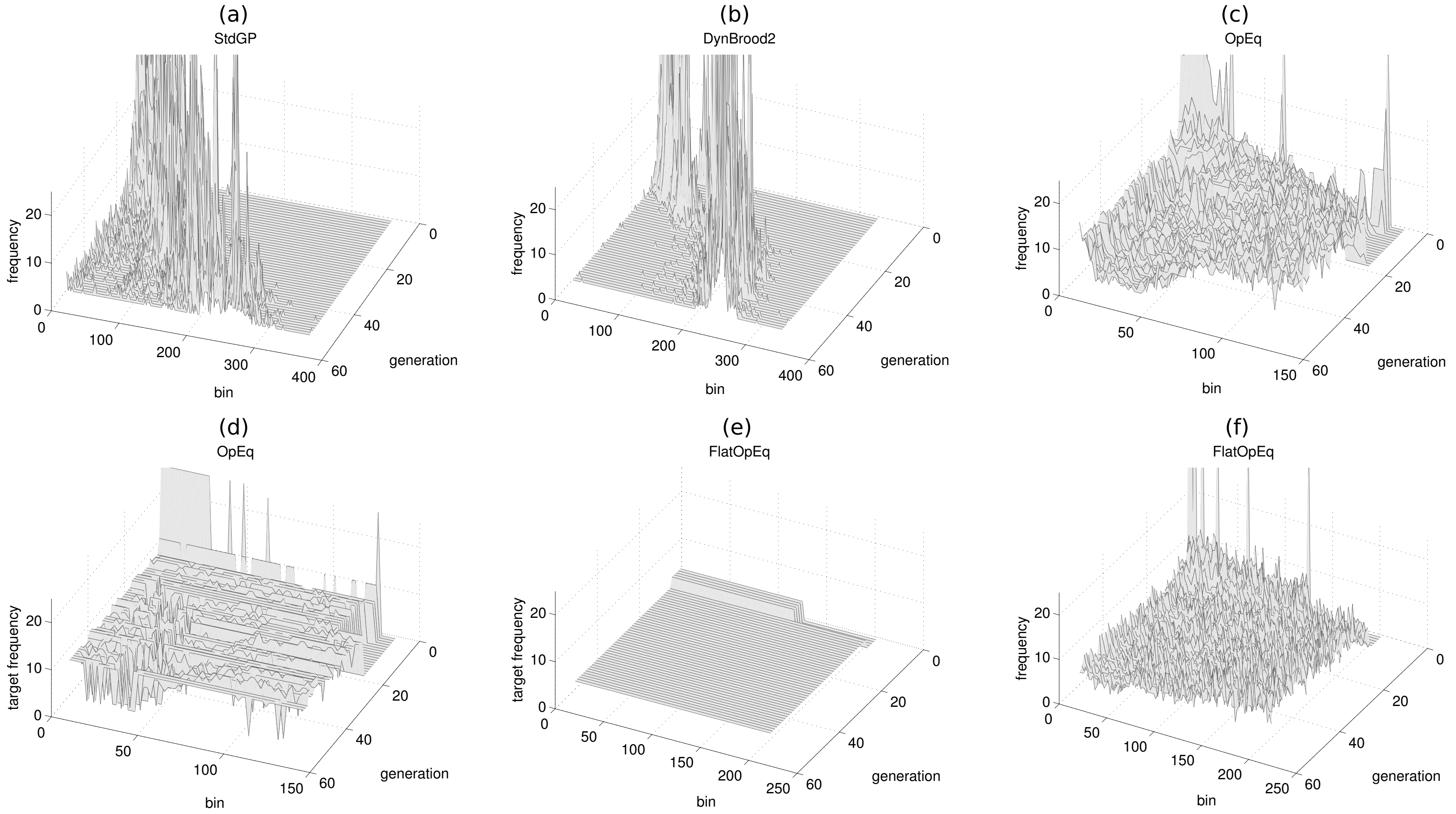


Fig. 3: Examples of actual and target length distributions. (a) Typical length distribution of Standard GP; (b) Typical length distribution of DynBrood2; (c) Typical length distribution of Operator Equalisation; (d) Target length distribution that originated the distribution in plot c; (e) Typical target length distribution of Operator Equalisation with flat target; (f) Length distribution originated by the target in plot e. All frequencies above 25 are not shown.

There seems to be no substantial difference in terms of the larger individuals, at least nothing typical was evident among the 20 runs. However, in terms of the smaller individuals the effect of Brood Recombination can be clearly seen. While Standard GP keeps producing (and consequently accepting) small individuals long after the distribution is centered on larger lengths, Brood Recombination has the ability to prevent them from entering the population, exhibiting an almost clean cut between zero frequency and high frequency. This effect is stronger as the brood size increases, and further so when using Batch instead of Brood (not shown). We naturally assume that, given the similarity of both distributions, like Standard GP Brood Recombination is also producing small individuals. Therefore we conclude that they are indeed the most unfit ones and are thus rejected in favor of the best of the brood. According to the Crossover Bias theory (see Section 1), this should act against bloat, but our results show only a minimal effect, at least when compared to Operator Equalisation.

The length distribution of Operator Equalisation (Figure 3c) is completely different from the previous ones. Instead of showing a clear preference for given lengths, this distribution spreads the individuals across most of the available bins, including the bins of the smaller lengths that presumably contain the worst individuals. This surprising “flatness” was found to be the main characteristic of all the 20 distributions of Operator Equalisation. Further investigation revealed that these distributions are the result of following targets that are mostly uniform, a truly unexpected finding. The plot of Figure 3d shows the target that originated the distribution shown on plot c. Had the target been faithfully followed, the actual distribution would be even flatter, but the several rejections and overrides gave it a less artificial look.

The explanation for such a flat uniform target became clear after realizing that the diversity and amplitude of fitness values that occur in this real world problem almost guarantees the presence of very unfit outliers in the population, practically every generation. These fitness values are so much worse than the others that, compared to them, all the rest looks the same, and all the bins end up getting the same capacity. Removing these outliers before calculating the target may prove to be a difficult task, since once we remove the first lot others will appear in the new distribution. An obvious improvement is to use the best or median fitness of each bin, instead of the average, to calculate the target, but even so the problem persists when some bins contain only very unfit individuals.

But is this really a problem that needs to be solved? Certainly an unintended feature, but also the most probable reason why Operator Equalisation exhibits a behavior so different and so much better than all other techniques, at least where bloat is concerned. No matter where the best individuals are found, Operator Equalisation maintains an almost uniform search across the entire set of explored lengths, thus increasing the probability of finding smaller solutions. In the limit, the number of individuals in the population will not be enough to ensure one individual per bin, but let us concentrate on the most immediate issues for now.

6.3 Enforcing a flat target

Assuming the nearly flat length distribution target is the true responsible for the success of OpEq in this symbolic regression problem, we wonder what improvements we can achieve if we enforce a truly flat distribution. So we implemented an Operator Equalisation variant that does not calculate the capacity of the bins with Equation 2, but instead gives the same capacity to all of them. All the rest, in particular the decision to create new bins, did not suffer any changes. Note that for most problems the actual length distribution is never exactly equal to the target, because when bins get full the target begins to be overridden. In our particular problem this is aggravated by the fact that all the arithmetic operators in the function set are binary, making it impossible to create solutions of even length. This means that half of the bins of our perfectly flat target are never filled, and half or the individuals of the population are guaranteed to override the target. In fact, because of this limitation caused by the set of exclusively binary functions, to facilitate the visualization none of the plots of Figure 3 shows the bins of even length.

Figure 3e shows a typical target distribution of the new variant of Operator Equalisation, that we call FlatOpEq, while Figure 3f shows the actual distribution obtained by using this target. It is not completely flat for the reasons stated above, but it is typically much flatter than the actual distributions of OpEq, like the one in Figure 3c. Next, we compare FlatOpEq with the remaining techniques, in terms of fitness and solution length.

Figure 4 shows a direct comparison between Standard GP with and without Dynamic Limits, Operator Equalisation with and without flat target, and another choice of a Brood/Batch technique, in this case Batch5 for being the one with the more desirable bloating behavior among all the Brood/Batch variants, with or without Dynamic Limits.

Figure 5 shows boxplots of the training and test fitness values, and the solution lengths, obtained in the final generations of the 20 runs. The FlatOpEq technique can reach significantly better fitness values than all other techniques (determined by non-parametric ANOVA with $p = 0.05$), but once again the explanation may simply be that more rejections mean more search, hence more learning. In terms of test fitness there are no statistically significant differences (Figure 5b), however it is worth noting that somewhere along the evolution FlatOpEq is able to reach better test fitness than the other techniques, before overfitting occurs (Figure 4b). The average solution length is basically the same for both Operator Equalisation variants at the end of the run, for despite doing more search FlatOpEq stabilizes the average solution length at around the same values as OpEq.

All comparisons made, there seems to be no disadvantage in artificially flattening the target distribution of Operator Equalisation.

7 Conclusions and Future Work

We have hypothesized that the two key ingredients that make Operator Equalisation such a successful bloat control method can be found in older methods such as Brood Recombination and Dynamic Limits. With Brood Recombination, the usually very unfit small individuals frequently produced by the parents are rejected in favor of the best of the brood. According to the Crossover Bias theory, eliminating the bias to introduce small unfit individuals in the population helps control bloat. With Dynamic Limits, the individuals larger than any others already found in the population are only accepted if they prove to be the best ever found during the run. This prevents unnecessarily large individuals to enter the population, thus controlling bloat.

We reassembled Operator Equalisation by taking a Standard GP system and coupling it with these two ingredients, obtaining a hybrid method which we tested in a hard real world regression problem. None of the several variants tested was able to produce a bloating behavior remotely similar to the one of Operator Equalisation. We took a deeper look at the dynamics of the search and found that, for previously unsuspected reasons, the target length distribution used by Operator Equalisation is typically nearly flat, contrasting with the peaky and well delimited targets of all the other approaches. Finally we introduced a new Operator Equalisation variant that enforces an artificially created flat target, and verified that the results were even better than the previous version.

It seems like the flatter the target, the most success is achieved in bloat control. Instead of avoiding small unfit individuals, the flat target actually prevents the search from moving away from the shorter lengths, even long after better and larger solutions have been found. It simply spreads individuals across all the previously visited lengths, ensuring that search does not abandon any of them.

After absorbing these results it becomes quite trivial that, to avoid bloat and reach smaller solutions, we must keep searching among the shorter lengths. The success of Operator Equalisation is undeniable, but the current results force us to look back at its previous successes and check if they were simply the result of an unintended flat distribution target, or if the Crossover Bias theory actually plays a significant role in the process. We realize just now that a flat target may appear as a consequence of, not only extremely high, but also extremely low, phenotypic diversity, and the benchmark parity problems immediately come to mind as cases to check. We leave this as future work. We also intend to provide results based on some measure of computational effort, for example the number of evaluations performed, instead of the number of generations, to make the comparison between techniques more objective and fair.

We finish with the ironic remark that the original meaning of equalization was, not surprisingly, flattening the signal along the entire spectrum.

Acknowledgments

This work was partially supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. The author thanks project PTDC/EIA-CCO/103363/2008 from FCT, Portugal. Thank you also to the anonymous reviewers for all their helpful comments.

References

[1] L. Altenberg. The evolution of evolvability in genetic programming. In K. E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.

[2] F. Archetti, S. Lanzeni, E. Messina, and L. Vanneschi. Genetic programming for human oral bioavailability of drugs. In M. Keijzer, et al., editors, *GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, volume 1, pages 255–262, Seattle, Washington, USA, 8-12 July 2006. ACM Press.

[3] F. Archetti, S. Lanzeni, E. Messina, and L. Vanneschi. Genetic programming for computational pharmacokinetics in drug discovery and development. *Genetic Programming and Evolvable Machines*, 8(4):413–432, Dec. 2007. special issue on medical applications of Genetic and Evolutionary Computation.

[4] M. Brameier and W. Banzhaf. Neutral variations cause bloat in linear GP. In C. Ryan, et al., editors, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 286–296, Essex, 14-16 Apr. 2003. Springer-Verlag.

[5] S. Dignum and R. Poli. Generalisation of the limiting distribution of program sizes in tree-based genetic programming and analysis of its effects on bloat. In D. Thierens, et al., editors, *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, volume 2, pages 1588–1595, London, 7-11 July 2007. ACM Press.

[6] S. Dignum and R. Poli. Crossover, sampling, bloat and the harmful effects of size limits. In M. O'Neill, et al., editors, *Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008*, volume 4971 of *Lecture Notes in Computer Science*, pages 158–169, Naples, 26-28 Mar. 2008. Springer.

[7] S. Dignum and R. Poli. Operator equalisation and bloat free GP. In M. O'Neill, et al., editors, *Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008*, volume 4971 of *Lecture Notes in Computer Science*, pages 110–121, Naples, 26-28 Mar. 2008. Springer.

[8] S. Gelly, O. Teytaud, N. Bredeche, and M. Schoenauer. Universal consistency and bloat in GP. *Revue d'Intelligence Artificielle*, 20(6):805–827, 2006. Issue on New Methods in Machine Learning. Theory and Applications.

[9] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[10] N. F. McPhee and J. D. Miller. Accurate replication in genetic programming. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)*, pages 303–309, Pittsburgh, PA, USA, 15-19 July 1995. Morgan Kaufmann.

[11] R. Poli, W. B. Langdon, and S. Dignum. On the limiting distribution of program sizes in tree-based genetic programming. In M. Ebner, et al., editors, *Proceedings of the 10th European Conference on Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, pages 193–204, Valencia, Spain, 11-13 Apr. 2007. Springer.

[12] R. Poli, N. F. McPhee, and L. Vanneschi. Analysis of the effects of elitism on bloat in linear and tree-based genetic programming. In R. L. Riolo, et al., editors, *Genetic Programming Theory and Practice VI*, Genetic and Evolutionary Computation, chapter 7, pages 91–111. Springer, Ann Arbor, 15-17May 2008.

[13] R. Poli, N. F. McPhee, and L. Vanneschi. Elitism reduces bloat in genetic programming. In M. Keijzer, et al., editors, *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1343–1344, Atlanta, GA, USA, 12-16 July 2008. ACM.

[14] R. Poli, N. F. McPhee, and L. Vanneschi. The impact of population size on code growth in GP: analysis and empirical validation. In M. Keijzer, et al., editors, *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1275–1282, Atlanta, GA, USA, 12-16 July 2008. ACM.

[15] S. Silva and J. Almeida. Dynamic maximum tree depth. In E. Cantú-Paz, et al., editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1776–1787, Chicago, 12-16 July 2003. Springer-Verlag.

[16] S. Silva and E. Costa. Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. *Genetic Programming and Evolvable Machines*, 10(2):141–179, 2009.

[17] S. Silva and S. Dignum. Extending operator equalisation: Fitness based self adaptive length distribution for bloat free GP. In L. Vanneschi, et al., editors, *Proceedings of the 12th European Conference on Genetic Programming, EuroGP 2009*, volume 5481 of *LNCS*, pages 159–170, Tuebingen, Apr. 15-17 2009. Springer.

[18] S. Silva and L. Vanneschi. Operator equalisation, bloat and overfitting: a study on human oral bioavailability prediction. In G. Raidl, et al., editors, *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1115–1122, Montreal, 8-12 July 2009. ACM.

[19] S. Silva and L. Vanneschi. Bloat free genetic programming: Application to human oral bioavailability prediction. *International Journal of Data Mining and Bioinformatics*, to appear.

[20] S. Silva, M. Vasconcelos, and J. Melo. Bloat free genetic programming versus classification trees for identification of burned areas in satellite imagery. In C. Di Chio, et al., editors, *Applications of Evolutionary Computation: EvoApplications 2010: Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP)*, volume 6024 of *LNCS*, Istanbul, 7-9 Apr. 2010. Springer.

[21] W. A. Tackett. *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA, 1994.

[22] L. Vanneschi and S. Silva. Using operator equalisation for prediction of drug toxicity with genetic programming. In L. S. Lopes, et al., editors, *Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence, EPIA 2009*, volume 5816 of *LNAI*, pages 65–76, Aveiro, Portugal, Oct. 12-15 2009. Springer.

About the authors

Sara Silva is a senior researcher of the Knowledge Discovery and Bioinformatics (KDBIO) group at INESC-ID Lisboa, Portugal, and an invited researcher of the Evolutionary and Complex Systems (ECOS) group at CISUC, Portugal. She has a BSc (5 years, finished in 1996) and a MSc (finished in 1999) in Informatics by the Faculty of Sciences of the University of Lisbon, Portugal, and a PhD (finished in 2008) in Informatics Engineering by the Faculty of Sciences and Technology of the University of Coimbra, Portugal. After graduation she used neural networks and genetic algorithms in several interdisciplinary projects ranging from remote sensing and forest science to epidemiology and medical informatics. She started her research on Genetic Programming (GP) in 2002, studying the bloat problem. Her main contributions to this field were the Dynamic Limits and Resource-Limited GP bloat control methods, and the developments that put into practice the new Operator Equalisation method. Her current main research interests are bloat and overfitting in GP, and how they relate to each other, and the effective and efficient usage of GP in real life problems within the earth sciences and bioinformatics domains. She is a member of the editorial board of Genetic Programming and Evolvable Machines, and the creator and developer of GPLAB - A Genetic Programming Toolbox for MATLAB.

Homepage: http://kdbio.inesc-id.pt/~sara/
Email: sara@kdbio.inesc-id.pt

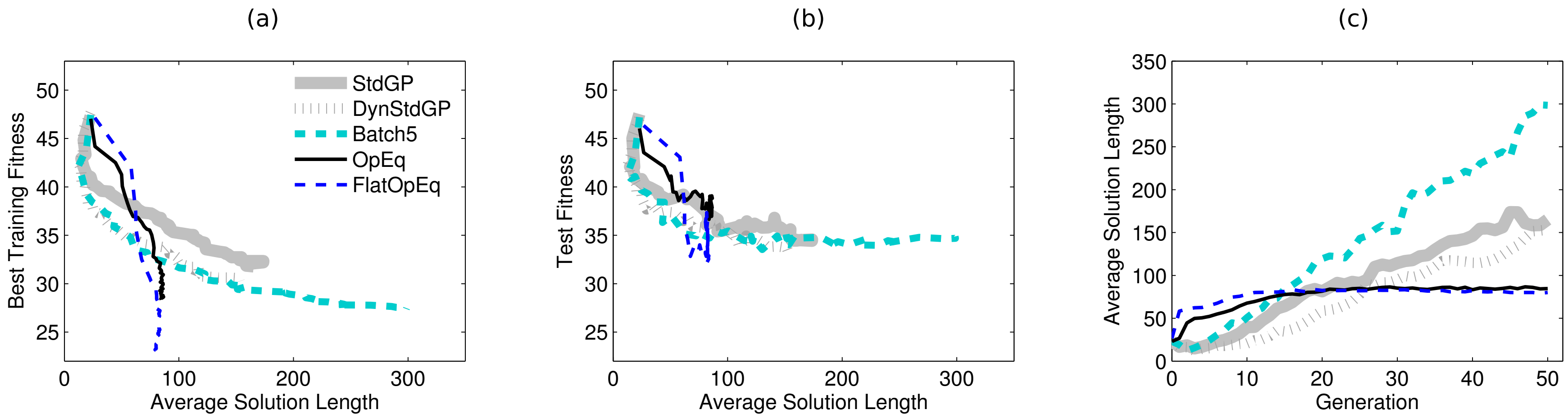


Fig. 4: Standard GP with and without Dynamic Limits versus one of the Brood/Batch techniques (Batch5) versus Operator Equalisation with and without flat target. (a) Best training fitness versus average solution length; (b) Test fitness (of the best training individual) versus average solution length; (c) Average solution length versus generations.

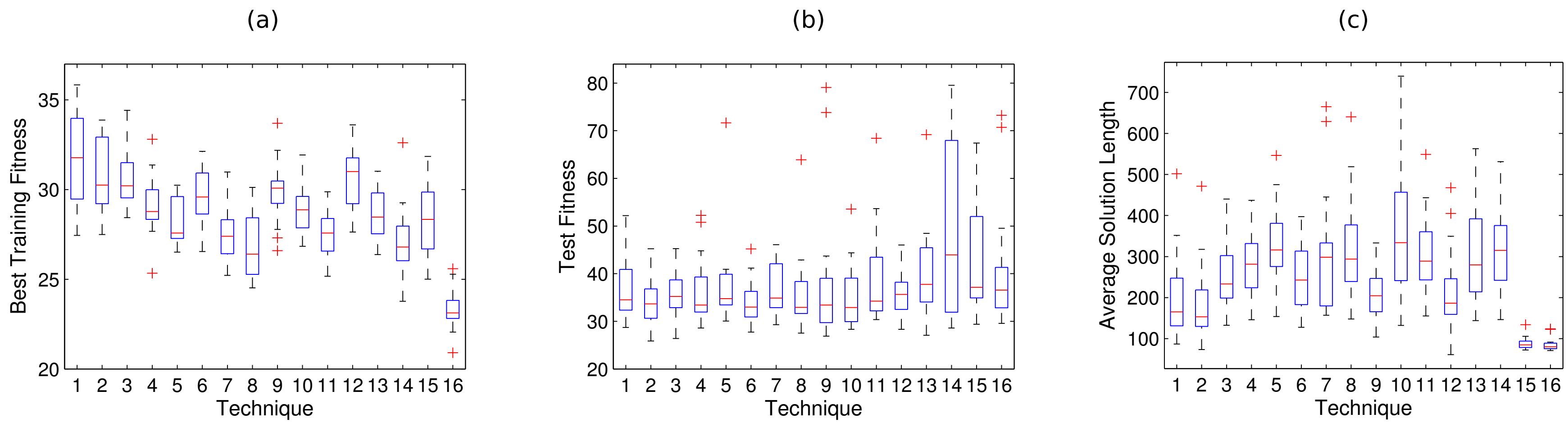
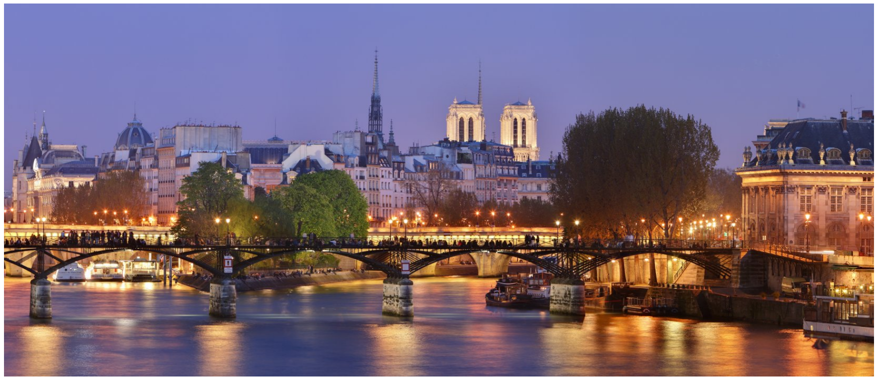


Fig. 5: Boxplots of all the 16 techniques used (see Table 1 for their acronyms). (a) Best training fitness; (b) Test fitness (of the best training individual); (c) Average solution length. Values obtained in the last generation of each of the 20 runs. Many outliers not shown in plot b.

Calls and Calendar

January 2012



Learning and Intelligent OptimizatioN Conference - LION 6

January 16-20, 2012, Paris, France

Homepage: http://www.intelligent-optimization.org/LION6/

Call for Papers: www

Deadline October 14, 2011

Notification to authors: November 28, 2011

Conference dates: January 16-20, 2012

Camera ready for post-proceedings: February 24, 2012

The LION conference aims at exploring the intersections between machine learning, artificial intelligence, mathematical programming and algorithms for hard optimization problems. The main purpose of the event is to bring together experts from all these areas to present and discuss new ideas, new methods, general trends, challenges and opportunities in applications as well as in research aiming at algorithmic advances. The conference program will consist of plenary presentations, introductory and advanced tutorials, technical presentations, and it will give ample time for discussions.

Relevant Research Areas

LION 6 solicits contributions dealing with all aspects of learning and intelligent optimization. Topics of interest include, but are not limited to:

- Metaheuristics such as tabu search, iterated local search, evolutionary algorithms, ant colony optimization, particle swarm optimization, and memetic algorithms
- Hybridizations of metaheuristics with other techniques for optimization
- Hyperheuristics and automatic design of heuristics
- Machine learning-aided search and optimization
- Algorithm portfolios and off-line tuning methods
- Reactive search optimization, autonomous search, adaptive and self-adaptive algorithms
- Specific adaptive metaheuristic techniques applied to propositional satisfiability, scheduling and planning, routing and logistics problems
- Interface(s) between discrete and continuous optimization
- Algorithms for dynamic, stochastic and multi-objective problems
- Multiscale and multilevel methods

For all the previous approaches:

- Experimental analysis and modeling
- Parallelization techniques
- Theoretical foundations
- Innovative applications

High-quality scientific contributions to these topics are solicited, in addition to advanced case studies from interesting, high-impact application areas.

Submission Details

LION 6 accepts the following three submission types:

- Long paper: original novel and unpublished work (max. 15 pages in Springer LNCS format);
- Short paper: an extended abstract of novel work (max. 4 pages in Springer LNCS format);
- Work for oral presentation only (in any Latex format, no page restriction). For example work already published elsewhere, which is relevant and which may solicit fruitful discussion at the conference.

Further Information

Up-to-date information will be published on the web site www.intelligent-optimization.org/LION6. For information about local arrangements, registration forms, etc., please refer to the above-mentioned web site or contact the organizers.

LION 6 Conference and Technical co-chairs

- Youssef Hamadi, Microsoft Research, UK (youssefh@microsoft.com)
- Marc Schoenauer, INRIA, France (Marc.Schoenauer@inria.com)

April 2012

Evostar 2012 - EuroGP, EvoCOP, EvoBIO, EvoMusart and EvoApplications
April 11-13, 2012, Malaga, Spain
Homepage: http://www.evostar.org
Flyer: pdf
Deadline November 30, 2011
Notification to authors: January 14, 2012
Camera-ready deadline: February 5, 2012

evo* comprises the premier co-located conferences in the field of Evolutionary Computing: **eurogp**, **evocop**, **evobio**, **evomusart** and **evoapplications**.

Featuring the latest in theoretical and applied research, evo* topics include recent genetic programming challenges, evolutionary and other meta-heuristic approaches for combinatorial optimization, evolutionary algorithms, machine learning and data mining techniques in the biosciences, in numerical optimization, in music and art domains, in image analysis and signal processing, in hardware optimization and in a wide range of applications to scientific, industrial, financial and other real-world problems.

eurogp (flyer)

15th European Conference on Genetic Programming Papers are sought on topics strongly related to the evolution of computer programs, ranging from theoretical work to innovative applications.

evocop (flyer)

12th European Conference on Evolutionary Computation in Combinatorial Optimization Practical and theoretical contributions are invited, related to evolutionary computation techniques and other meta-heuristics for solving combinatorial optimization problems.

evobio (flyer)

10th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology Emphasis is on evolutionary computation and other advanced techniques addressing important problems in molecular biology, proteomics, genomics and genetics, that have been implemented and tested in simulations and on real-life datasets.

evomusart (flyer)

1st International Conference and 10th European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design

evoapplications (flyer)

European Conference on the Applications of Evolutionary Computation

evocomnet

9th European event on nature-inspired techniques for telecommunication networks and other parallel and distributed systems

evocomplex

3rd European event on algorithms and complex systems

evofin

6th European event on evolutionary and natural computation in finance and economics

evogames

4th European event on bio-inspired algorithms in games

evohot

7th European event on bio-inspired heuristics for design automation

evoiasp

14th European event on evolutionary computation in image analysis and signal processing

evonum

5th European event on bio-inspired algorithms for continuous parameter optimisation

evopar

1st European event on parallel and distributed Infrastructures

evorisk

1st European event on computational intelligence for risk management, security and defence applications

evostim

7th European event on nature-inspired techniques in scheduling, planning and timetabling

evostoc

9th European event on evolutionary algorithms in stochastic and dynamic environments

evotranslog

6th European event on evolutionary computation in transportation and logistics

GECCO 2012 - Genetic and Evolutionary Computation Conference

July 7-11, 2012, Philadelphia, PA, USA

Homepage: http://www.sigevo.org/gecco-2012

Deadline January 13, 2012

Author notification: March 13, 2012

Workshop and tutorial proposals submission: November 07, 2011

Notification of workshop and tutorial acceptance: November 28, 2011

The Genetic and Evolutionary Computation Conference (GECCO-2012) will present the latest high-quality results in the growing field of genetic and evolutionary computation.

Topics include: genetic algorithms, genetic programming, evolution strategies, evolutionary programming, real-world applications, learning classifier systems and other genetics-based machine learning, evolvable hardware, artificial life, adaptive behavior, ant colony optimization, swarm intelligence, biological applications, evolutionary robotics, coevolution, artificial immune systems, and more.

Organizers

General Chair:	Jason Moore
Editor-in-Chief:	Terence Soule
Publicity Chair:	Xavier Llorá
Tutorials Chair:	Gabriela Ochoa
Students Chair:	Josh Bongard
Workshops Chair:	Bill Rand
Competitions Chairs:	Daniele Loiacono
Business Committee:	Wolfgang Banzhaf Marc Schoenauer
EC in Practice Chairs:	Jörn Mehnen Thomas Bartz-Beielstein, David Davis

Important Dates

Paper Submission Deadline	January 13, 2012
Decision Notification	March 13, 2012
Camera-ready Submission	April 9, 2012

To Propose a Tutorial or Workshop

A detailed call for workhop and tutorial proposals will be posted later so stay tuned! Meanwhile, for enquiries regarding tutorials contact gecco2012tutorials@sigevolution.org while for enquiries about workshops contact gecco2012workshops@sigevolution.org.

More Information

Visit www.sigevo.org/gecco-2012 for information about electronic submission procedures, formatting details, student travel grants, the latest list of tutorials and workshop, late-breaking papers, and more.

Contact

For general help and administrative matters contact GECCO support at gecco2012@sigevolution.org

GECCO is sponsored by the Association for Computing Machinery Special Interest Group for Genetic and Evolutionary Computation.

September 2012

PPSN 2012 – International Conference on Parallel Problem Solving From Nature
September 1-5, 2012, Taormina, Italy
Homepage: http://www.dmi.unict.it/ppsn2012/
Call for paper: www
Email: ppsn2012@dmi.unict.it
Paper Submission Deadline: March 15, 2012
Author Notification: June 1, 2012
Workshop Proposals Submission: October 15, 2011

PPSN XII will showcase a wide range of topics in Natural Computing including, but not restricted to: Evolutionary Computation, Quantum Computation, Molecular Computation, Neural Computation, Artificial Life, Swarm Intelligence, Artificial Ant Systems, Artificial Immune Systems, Self-Organizing Systems, Emergent Behaviors, and Applications to Real-World Problems.

Paper Presentation

Following the now well-established tradition of PPSN conferences, all accepted papers will be presented during small poster sessions of about 16 papers. Each session will contain papers from a wide variety of topics, and will begin by a plenary quick overview of all papers in that session by a major researcher in the field. Past experiences have shown that such presentation format led to more interactions between participants and to a deeper understanding of the papers. All accepted papers will be published in the LNCS Proceedings.

Paper Submission

Researchers are invited to submit original work in the field of natural computing as papers of not more than 10 pages. Authors are encouraged to submit their papers in LaTeX. Papers must be submitted in Springer Verlag's LNCS style through the conference homepage, here.

IEEE Conference on Computational Intelligence and Games (CIG-2012)
September 12-15, 2012, Granada, Spain
Homepage: http://geneura.ugr.es/cig2012/
Flyer: pdf
Submission deadline: April 15, 2012
Decision notification: June 1, 2012
Camera-ready submission: June 15, 2012
Conference: September 12-15, 2012

Aim and Scope

Games have proven to be an ideal domain for the study of computational intelligence as not only are they fun to play and interesting to observe, but they provide competitive and dynamic environments that model many real-world problems. Additionally, methods from computational intelligence promise to have a big impact on game technology and development, assisting designers and developers and enabling new types of computer games. The 2010 IEEE Conference on Computational Intelligence and Games brings together leading researchers and practitioners from academia and industry to discuss recent advances and explore future directions in this quickly moving field.

Topics of interest include, but are not limited to:

- Learning in games
- Coevolution in games
- Neural-based approaches for games
- Fuzzy-based approaches for games
- Player/Opponent modeling in games
- CI/AI-based game design
- Multi-agent and multi-strategy learning
- Applications of game theory
- CI for Player Affective Modeling
- Intelligent Interactive Narrative
- Imperfect information and non-deterministic games
- Player satisfaction and experience in games
- Theoretical or empirical analysis of CI techniques for games
- Comparative studies and game-based benchmarking
- Computational and artificial intelligence in:
    - Video games
    - Board and card games
    - Economic or mathematical games
    - Serious games
    - Augmented and mixed-reality games
    - Games for mobile platforms

The conference will consist of a single track of oral presentations, tutorial and workshop/special sessions, and live competitions. The proceedings will be placed in IEEE Xplore, and made freely available on the conference website after the conference.

Conference Committee

General Chair:	Antonio J. Fernández Leiva
Program Chairs:	Simon Lucas, Sung-Bae Cho, and Magy Seif El-Nasr
Publicity Chair:	Antonio M. Mora García
Social Media Chair:	Juan J. Merelo
Finance Chair:	Pedro A. Castillo
Proceedings Chairs:	Mike Preuss and Anna I. Esparcia
Competition Chair:	Julian Togelius
Special Sessions and Tutorials Chair:	Georgios Yannakakis
Local Chairs:	Carlos Cotta Porras, Antonio J. Fernández Leiva, Antonio M. Mora García, Juan J. Merelo, and Pedro A. Castillo

Important Dates

Tutorial proposals:	15 March 2012
Paper submission:	15 April 2012
Decision Notification:	1 June 2012
Camera-ready:	15 June 2012
Conference:	12-15 September 2012

For more information please visit: http://geneura.ugr.es/cig2012/

About the Newsletter

SIGEVOlution is the newsletter of SIGEVO, the ACM Special Interest Group on Genetic and Evolutionary Computation.

To join SIGEVO, please follow this link [WWW]

Contributing to SIGEVOlution

We solicit contributions in the following categories:

Art: Are you working with Evolutionary Art? We are always looking for nice evolutionary art for the cover page of the newsletter.

Short surveys and position papers: We invite short surveys and position papers in EC and EC related areas. We are also interested in applications of EC technologies that have solved interesting and important problems.

Software: Are you are a developer of an EC software and you wish to tell us about it? Then, send us a short summary or a short tutorial of your software.

Lost Gems: Did you read an interesting EC paper that, in your opinion, did not receive enough attention or should be rediscovered? Then send us a page about it.

Dissertations: We invite short summaries, around a page, of theses in EC-related areas that have been recently discussed and are available online.

Meetings Reports: Did you participate in an interesting EC-related event? Would you be willing to tell us about it? Then, send us a short summary, around half a page, about the event.

Forthcoming Events: If you have an EC event you wish to announce, this is the place.

News and Announcements: Is there anything you wish to announce? This is the place.

Letters: If you want to ask or to say something to SIGEVO members, please write us a letter!

Suggestions: If you have a suggestion about how to improve the newsletter, please send us an email.

Contributions will be reviewed by members of the newsletter board.

We accept contributions in LaTeX, MS Word, and plain text.

Enquiries about submissions and contributions can be emailed to editor@sigevolution.org.

All the issues of SIGEVOlution are also available online at www.sigevolution.org.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in the Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.